

Физический факультет МГУ им. М.В.Ломоносова  
Курсовая работа  
по программированию

Оптимизация расчетных сеток для векторных  
дифракционных интегралов в задачах жесткой  
фокусировки лазерного излучения

Выполнил студент  
2 курса  
Долгих Константин Александрович  
Научный руководитель: проф. Косарева О.Г.

Москва 2017

## Содержание

Введение.....	3
Теория.....	4
Методы.....	7
Результаты и анализ.....	12
Выводы.....	18
Список литературы.....	19

## Введение

Сейчас лазерные системы позволяют получать импульсы мощностью  $\sim 1$  ПВт. Их фокусировка в пятно с размерами порядка длины позволяет достичь интенсивности  $10^{22-24}$  Вт/см<sup>2</sup>. При таких полях заряженные частицы приобретают очень большое ускорение, разгоняясь до релятивистских скоростей, и материя оказывается в экзотическом состоянии, что используется в изучении явлений нелинейной оптики и других экспериментах. При длине волны лазерного излучения около 800 нм пучок можно сфокусировать в пятно порядка длины волны (около 900 нм). При таких условиях очень важно как можно точнее предсказывать поле в области фокуса [1, 2, 3].

Некоторые обозначения, используемые в работе:  $NA = \sin(\theta)$ , где  $2\theta$  – угол, под которым границы пучка сходятся в фокусе зеркала. f-number,  $f\# = 1/(2NA)$  – один из параметров фокусировки, равный в параксиальном приближении отношению  $f/d$ , где  $f$  – фокусное расстояние,  $d$  – диаметр фокусируемого пучка.

Для расчёта поля используются векторные интегралы Френеля-Кирхгофа, которые при достаточно широких пучках и малых значениях параметра фокусировки  $f\#$  – очень ресурсоёмкая задача: т. к. интеграл вычисляется по поверхности, площадь которой пропорциональна квадрату радиуса пучка, время выполнения программы тоже пропорционально квадрату радиуса, а так же обратно пропорционально шагу разбиения. При слишком большом шаге разбиения интеграл вычисляется некорректно. Уменьшать шаг разбиения тоже нельзя, потому что при этом увеличивается сложность, а, следовательно, и время вычислений. При реальных параметрах установки (радиусе пучка порядка сантиметра или более,  $f\#$  порядка единицы) сложность вычислений возрастает настолько, что поиск оптимального шага разбиения становится актуальной и важной задачей.

Один из возможных вариантов оптимизации расчётов (распараллеливание алгоритма и оптимальный выбор диаметра разбиения) описывается в данной работе, при этом рассматривалась только X-компонента напряжённости поля при условии фокусировки ( $1 < f\# < 10$ ).

## Теория.

**Принцип Гюйгенса-Френеля.** Основополагающим принципом волновой теории является принцип Гюйгенса-Френеля, который гласит, что каждый элемент волнового фронта можно рассматривать как центр вторичного возмущения, порождающего вторичные сферические волны, а результирующее световое поле в каждой точке пространства будет определяться интерференцией этих волн (рис.1) [4]. Исходя из этого принципа поле в точке, удалённой от некоторой излучающей поверхности (которая может быть поверхностью зеркала или отверстием в экране), можно описать интегралом по поверхности, где подынтегральное выражение описывает поле от одной точки на этой поверхности:

$$E(\vec{r}) = \frac{C}{r} e^{i(\omega t - kr)} = A e^{i\omega t},$$

где  $C$  – некоторая константа,  $A$  – скалярная комплексная амплитуда,  $k$  – волновое число,  $\omega$  – частота,  $t$  – момент времени. В данной работе мы будем рассматривать случай монохроматического излучения и опустим экспоненту  $\exp(i\omega t)$  в дальнейших выкладках. Пример использования: пусть есть точечный источник света, между точкой наблюдения и источником находится непрозрачный экран с вырезанной поверхностью  $\Sigma$ . Тогда амплитуду поля в точке наблюдения можно описать интегралом:

$$U(P) = \iint_{\Sigma} \frac{C e^{ik(\rho_1 + \rho)}}{\rho_1 \rho} \frac{i}{2\lambda} (\cos \varphi_1 + \cos \varphi) d\sigma$$

Это выражение называется дифракционной формулой Френеля-Кирхгофа [4]. При более точных расчётах следует использовать векторные дифракционные интегралы, учитывающие все три компоненты поля.

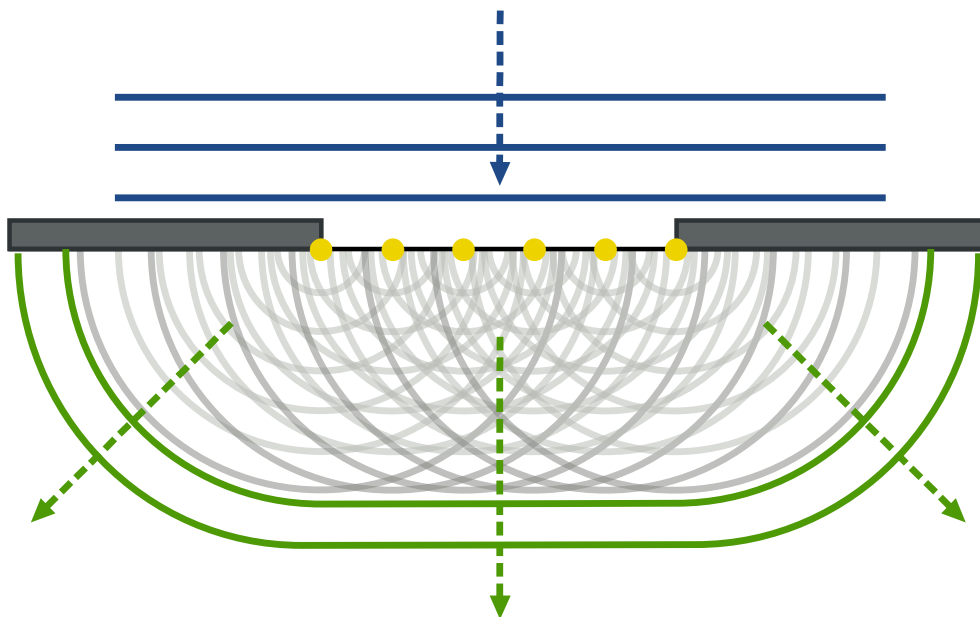


Рис. 1. Иллюстрация принципа Гюйгенса-Френеля  
<https://commons.wikimedia.org/w/index.php?curid=1944668>

**Параболическое зеркало.** Параболическое зеркало (геометрическая фигура – параболоид вращения) позволяет фокусировать параллельный пучок лучей в пятно размерами порядка длины волны. Как показано в [1], при условии жёсткой фокусировки это пятно можно уменьшить до  $\sim 0.9\lambda$ . При этом вокруг пятна в фокальной плоскости наблюдается ещё несколько колец (рис. 2, мой расчёт).

### Поле в фокальной плоскости зеркала

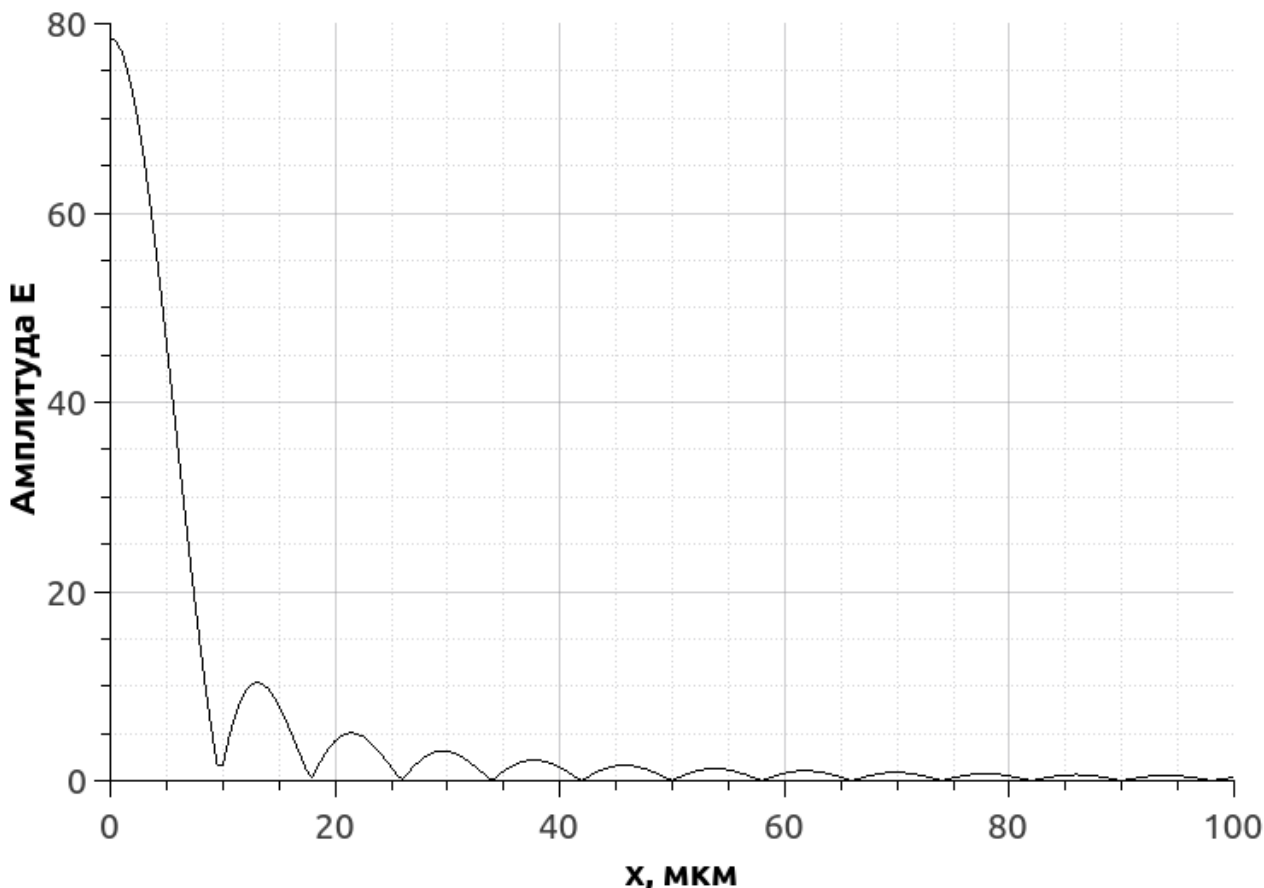


Рис. 2 Зависимость поля в области фокуса от смещения по фокальной плоскости

Точное значение  $X, Y, Z$ -составляющих поля в зависимости от координаты и других параметров описывается векторным дифракционным интегралом (выше приводился скалярный дифракционный интеграл, который рассматривается в курсе общей физики) [1,2,3,4]:

$$E_x(x_p, y_p, z_p) \sim \int_{\theta_m}^{\pi} \int_0^{2\pi} d\theta_s d\phi_s E_{inc}(\theta_s, \phi_s) \exp\{-ik\varphi(x_p, y_p, z_p, \theta_s, \phi_s)\} \frac{2f \sin \theta_s}{(1 - \cos \theta_s)},$$

$$\times \left\{ 1 - \frac{\sin \theta_s \cos \phi_s}{1 - \cos \theta_s} \left( 1 - \frac{1 - \cos \theta_s}{i2kf} \right) \frac{2f \sin \theta_s \cos \phi_s - x_p (1 - \cos \theta_s)}{2f} \right\}$$

$$E_y(x_p, y_p, z_p) \sim \int_{\theta_m}^{\pi} \int_0^{2\pi} d\theta_s d\phi_s E_{inc}(\theta_s, \phi_s) \exp\{-i\varphi(x_p, y_p, z_p, \theta_s, \phi_s)\} \frac{2f \sin \theta_s}{(1 - \cos \theta_s)^2},$$

$$\times \left\{ \sin \theta_s \cos \phi_s \left( 1 - \frac{1 - \cos \theta_s}{i2kf} \right) \frac{2f \sin \theta_s \sin \phi_s - y_p (1 - \cos \theta_s)}{2f} \right\}$$

$$E_z(x_p, y_p, z_p) \sim \int_{\theta_m}^{\pi} \int_0^{2\pi} d\theta_s d\phi_s E_{inc}(\theta_s, \phi_s) \exp\{-i\varphi(x_p, y_p, z_p, \theta_s, \phi_s)\} \frac{2f \sin \theta_s}{(1 - \cos \theta_s)^2},$$

$$\times \sin \theta_s \cos \phi_s \left\{ 1 - \left( 1 - \frac{1 - \cos \theta_s}{i2kf} \right) \frac{2f \cos \theta_s - z_p (1 - \cos \theta_s)}{2f} \right\}$$

$$\text{и } \varphi(x_p, y_p, z_p, \theta_s, \phi_s) = k(z_p \cos \theta_s + x_p \sin \theta_s \cos \phi_s + y_p \sin \theta_s \sin \phi_s) \quad [2]$$

В данной работе рассматривается упрощённый случай – только X-компонента напряжённости электрического поля:

$$\vec{E}(P) = -\vec{e}_x \frac{k}{4\pi} \exp(ikf) \iint dx_s dy_s \frac{\exp ik(u - \rho_s)}{u} E_i(x_s, y_s),$$

где  $\rho_s = z_s + 2f = \frac{x_s^2 + y_s^2}{4f} + f = \frac{x_s^2 + y_s^2 + 4f^2}{4f}$ ,  $u = |\vec{\rho}_s - \vec{\rho}_p|$ ,  $\rho_s$  – радиус-вектор точки на поверхности зеркала,  $\rho_p$  – радиус-вектор точки наблюдения,  $f$  – фокусное расстояние зеркала.

Этот интеграл не выражается в элементарных функциях, поэтому вычислять его следует численными методами [1,2].

**Теорема Котельникова** — фундаментальное утверждение в области цифровой обработки сигналов, связывающее непрерывные и дискретные сигналы и гласящее, что «любую функцию  $F(t)$ , состоящую из частот от 0 до  $f_1$ , можно непрерывно передавать с любой точностью при помощи чисел, следующих друг за другом через  $1/(2f_1)$  секунд»[3]. В данной работе проанализированные ошибки численного интегрирования возникают из-за эффекта отражения частот, являющегося следствием теоремы Котельникова.

## Методы.

**Методы вычисления кратных интегралов на компьютере.** Известно множество способов численного расчёта кратных интегралов на компьютере. Один из простейших – аналог метода трапеций для многомерного случая. Такой метод называется методом ячеек.

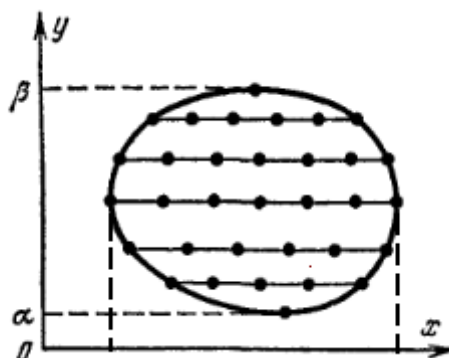


Рис. 3.

Рассмотрим двукратный интеграл по прямоугольнику  $G\{a < x < b, \alpha < y < \beta\}$  (рис. 3). По аналогии с формулой средних можно приближенно заменить функцию на ее значение в центральной точке прямоугольника. Тогда сам интеграл примерно равен интегральной сумме:

$$\int_a^b dx \int_{\alpha}^{\beta} dy f(x, y) \approx \sum_i \Delta x_i \Delta y_i f(\bar{x}_i, \bar{y}_i),$$

где  $\Delta x_i, \Delta y_i$  – стороны прямоугольников разбиения,  $\bar{x}_i, \bar{y}_i$  с чертой – средние точки прямоугольников разбиения. Для любой непрерывной на прямоугольнике  $G$  функции интегральная сумма будет сходиться к значению интеграла.

Ошибку вычисления интеграла можно оценить вычислив разницу между правой и левой частями формулы для вычисления интеграла [5]:

$$R = \int_a^b dx \int_{\alpha}^{\beta} dy f(x, y) - \sum_i \Delta x_i \Delta y_i f(\bar{x}_i, \bar{y}_i) \leq \frac{1}{24} (b-a)(\beta-\alpha) \left( \left( \frac{b-a}{N} \right)^2 \max f_{xx} + \left( \frac{\beta-\alpha}{M} \right)^2 \max f_{yy} \right)$$

где  $N$  и  $M$  – число точек по заданной оси. Формула даёт завышенную оценку ошибки, для более точной оценки следует заменить максимумы вторых производных на их среднее значение. Тем не менее видно, что ошибка убывает обратно пропорционально квадрату числа точек.

В данной работе погрешность вычислений не оценивалась, т.к. качественная оценка погрешности по формуле выше затруднительна, а корректность результата видна при анализе графика визуально. Тем не менее, в коде программы оставлена возможность вычисления погрешности: для этого необходимо посчитать сумму вторых производных подынтегральных функций на заданной области и подставить. Программа имеет в себе преобразованную для данного частного случая формулу вычисления ошибки.

**Технологии параллельного программирования.** На сегодняшний день процессоры почти достигли своей теоретически максимальной тактовой

частоты (при текущих технологиях). Пытаться увеличивать мощность ядра процессора не имеет смысла: устройство будет дорожать, а прибавка к производительности небольшая. Поэтому для ресурсоёмких вычислений следует использовать алгоритмы, допускающие распараллеливание, чтобы объединить производительность нескольких процессоров. Это позволяет существенно сократить время расчётов.

Оценим возможное ускорение при распараллеливании алгоритма на  $N$  процессах. Пусть  $t_0$  – время работы программы в последовательном режиме,  $t$  – в параллельном,  $a$  – последовательная часть алгоритма,  $b$  – часть, допускающая идеальное распараллеливание. Тогда  $t_0 = a + b$ ,  $t = a + b/N$ . А ускорение тогда равно:

$$\frac{t_0}{t} = \frac{a+b}{a+b/N} = \frac{1}{\alpha + \frac{1-\alpha}{N}}$$

где  $\alpha = a/(a+b)$  – последовательная доля алгоритма. Это соотношение называется законом Амдала [3]. Отсюда видно, что при  $N \rightarrow \infty$  ускорение не стремится к нулю. То есть, начиная с некоторого числа процессов, зависящего от доли последовательной части, ускорение будет меняться незначительно и дальнейшее распараллеливание не имеет смысла. В данной задаче алгоритм имеет очень малую последовательную долю, поэтому его имеет смысл распараллелить на почти неограниченном числе процессов.

Следует отметить, что модель закона Амдала не идеальна: здесь не учитывается, что при распараллеливании есть дополнительные временные затраты на создание параллельных процессов и/или обмен данными между ними.

Существует множество программных реализаций параллельного программирования. В моей программе реализовывались сразу две технологии параллельного программирования – OpenMP и MPI.

OpenMP – одна из самых простых в использовании технологий параллельного программирования. Плюсы OpenMP:

- Сравнительная простота написания кода
- Высокая скорость создания параллельных нитей
- При запуске программы обычно не требуется дополнительная настройка оборудования

Минусы:

- Невозможность реализовать распараллеливание между физически разделёнными процессорами (без общей памяти)
- В некоторых случаях простейший вариант распараллеливания не является оптимальным или даже не допустим, что трудно отлаживается.

MPI (Message Passing Interface) – технология параллельного программирования, позволяющая запускать программу на физически разных машинах (и даже под управлением разных ОС). Плюсы MPI:



- Позволяет объединять производительность множества машин с физически разделённой памятью, ядрами и пр. и даже под управлением различных операционных систем.

Минусы:

- Для оптимизации работы алгоритма программист должен более чётко понимать, что происходит в процессе работы программы, возможно, учитывая физическую архитектуру сети.
- Для настройки кластера с MPI необходим квалифицированный специалист и финансовые затраты для обеспечения быстрого обмена данными между узлами.

Так же существуют и другие технологии параллельного программирования (CUDA, OpenCV и др.) [1,6].

В данной работе использовались обе эти технологии: OpenMP и MPI. Программа вычисляет проекцию поля на ось X в множестве точек наблюдения с некоторым шагом. Вычисление поля для каждой точки – вычисление интеграла. Вычисление каждого отдельного интеграла распараллеливается с использованием OpenMP, а вычисление в различных точках – с использованием MPI. Такой подход удобен, потому что вычисление интеграла удобно производить с использованием общей памяти, а вычисление интеграла при различных параметрах – абсолютно не связанные друг с другом действия, которые можно распараллелить любым способом, но т.к. распараллеливание между ядрами внутри одного узла уже сделано при вычислении интеграла и программа должна быть универсальной, то здесь применяется распараллеливание между узлами. Такой подход позволяет максимально эффективно использовать ресурсы кластеров. Схема работы программы представлена на рисунке 4.

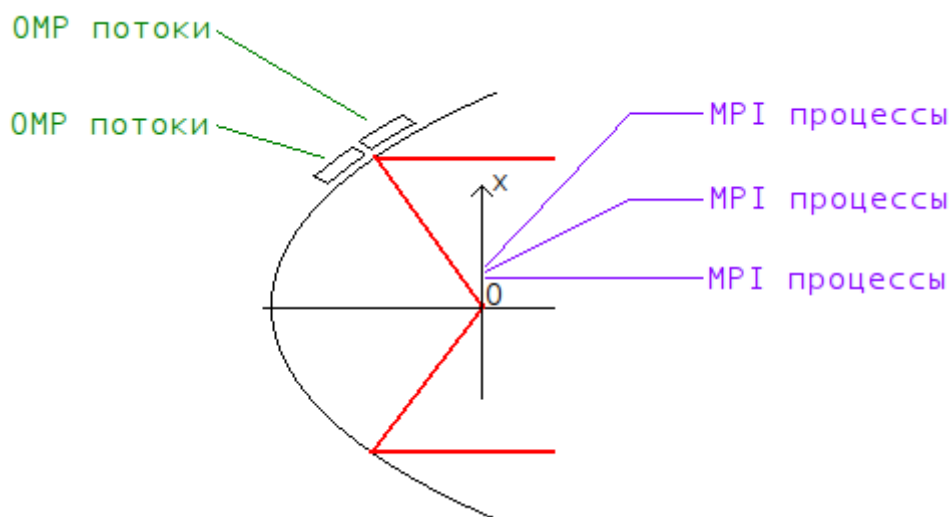


Рис. 4. Схема распараллеливания

### **Особенности вычисления интеграла от осциллирующей функции.**

Заметим, что подынтегральное выражение содержит экспоненту в мнимой

степени, следовательно, функция имеет некоторый период. Если при численном интегрировании шаг разбиения  $\Delta x_i$  и/или  $\Delta y_i$  совпадёт с периодом функции или близок к нему, то полученное значение интеграла будет значительно отличаться от истинного. На графике (рис.5) приводится пример плохого разбиения: здесь диаметры разбиения по 24 микрометра, при смещении от фокуса на 280-350 микрометров наблюдается несколько максимумов, которых с физической точки зрения быть не должно и на практике нет. Если диаметр разбиения уменьшать, то эти максимумы смещаются дальше от фокуса, расплываются, уменьшаются и исчезают, однако возрастает время вычислений.

### Расчёт поля в области фокуса при плохом разбиении

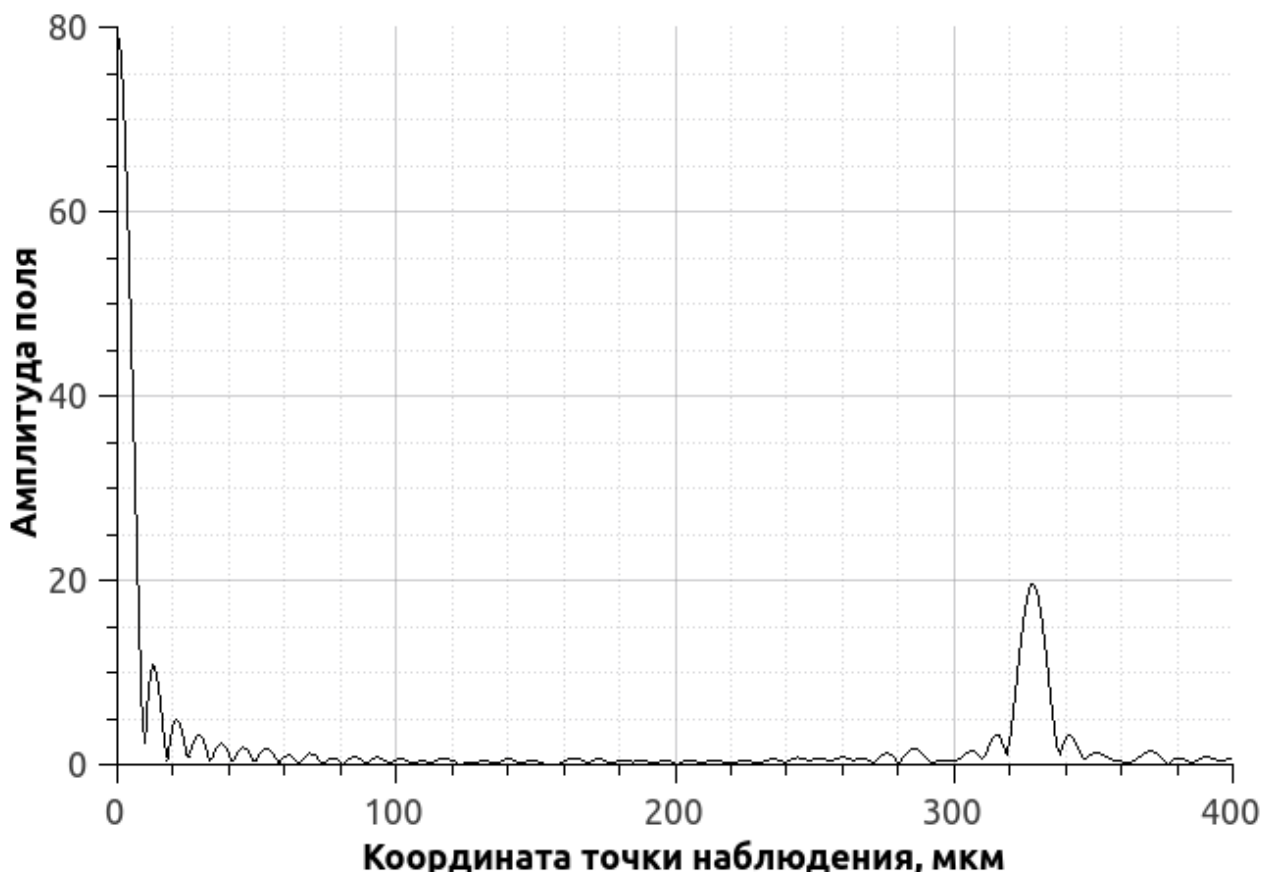


Рис. 5 Расчёт поля в фокусе при шаге разбиения 16.1 мкм. Максимум справа является ошибкой интегрирования.

Для оценки координаты появления артефактного максимума можно воспользоваться теоремой Котельникова заменив временную координату на пространственную. В данной задаче в качестве оси времени выступает координата на поверхности зеркала, а частота устройства – величина, обратная диаметру разбиения. Исходя из этих соображений получим связь между фокусным расстоянием, шагом разбиения и координатой, где интеграл считается неправильно.

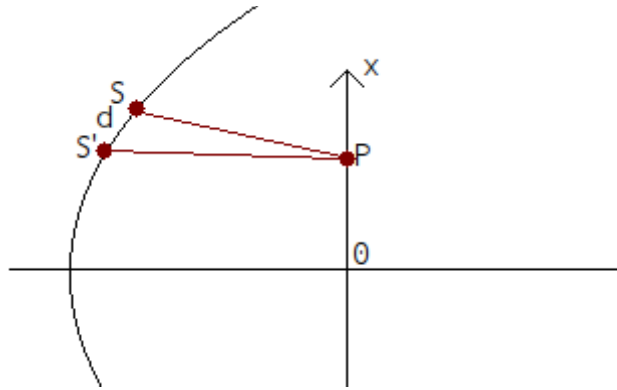


Рис. 6.

Пусть  $S_1$  – расстояние  $SP$ ,  $S_2$  – расстояние  $S'P$ ,  $X$ -координаты точек  $S$  и  $S'$  отличаются на шаг разбиения  $\Delta x$  (рис. 6). Для корректного расчёта интеграла по теореме Котельникова необходимо, чтобы  $|S_1 - S_2| < \lambda/2$  (для более точных оценок необходимо анализировать всю мнимую часть аргумента экспоненты, что технически очень трудоёмко). Вычислим эти расстояния:

$$S_1 = \frac{(X_s - X_p)^2}{4f} - f, \quad S_2 = \frac{((X_s - \Delta x) - X_p)^2}{4f} - f, \quad \text{где } f \text{ – фокусное расстояние.}$$

Тогда

$$|S_1 - S_2| = \left| \frac{(X_s - X_p)^2}{4f} - \frac{((X_s - \Delta x) - X_p)^2}{4f} \right| = \left| \frac{2X_s X_p - 2(X_s - \Delta x)X_p + (X_s - \Delta x)^2 - X_s^2}{4f} \right| \approx \frac{\Delta x (X_p - X_s)}{2f}$$

Отсюда можно оценить координату ошибочного вычисления интеграла

$$X_p \approx \frac{2f\lambda/2}{\Delta x} = \frac{2\pi f}{k\Delta x}. \quad \text{Видно, что } X_p \sim f \text{ и } X_p \sim 1/\Delta x.$$

## Результаты и анализ.

Программа написана на C++ 99 и протестирована на следующих многоядерных устройствах: домашний ноутбук (2 физических ядра), мобильный телефон (4 физических ядра), учебный тренировочный сервер (8 ядер) и СК «Ломоносов» (было использовано до 16 ядер и 16 узлов кластера) [5]. Хотя программа и реализует только случай параллельного пучка с функцией  $E(x,y)=\theta(R^2-x^2-y^2)$  (функция Хевисайда, модель идеального пучка) и только для одной компоненты напряжённости поля. Код легко модифицируем для более общих случаев (все три компоненты поля и пр.).

Программа стабильно работает и выдаёт корректный результат на всех устройствах и наблюдается сокращение времени её выполнения при увеличении числа процессов в соответствии с законом Амдала.

Результат теста программы при  $R=1,2$  мм на телефоне и ноутбуке на рисунке 7. Для чистоты эксперимента запускался абсолютно одинаковый код на обоих устройствах. Ввиду проблем с установкой пакета OpenMPI на смартфон, компоненты MPI были убраны. Как видно, 4-ядерный телефон (ARM Cortex-A53 988 МГц) считает заметно быстрее ноутбука с 2 ядрами (Intel Celeron N2840 2.16ГГц).

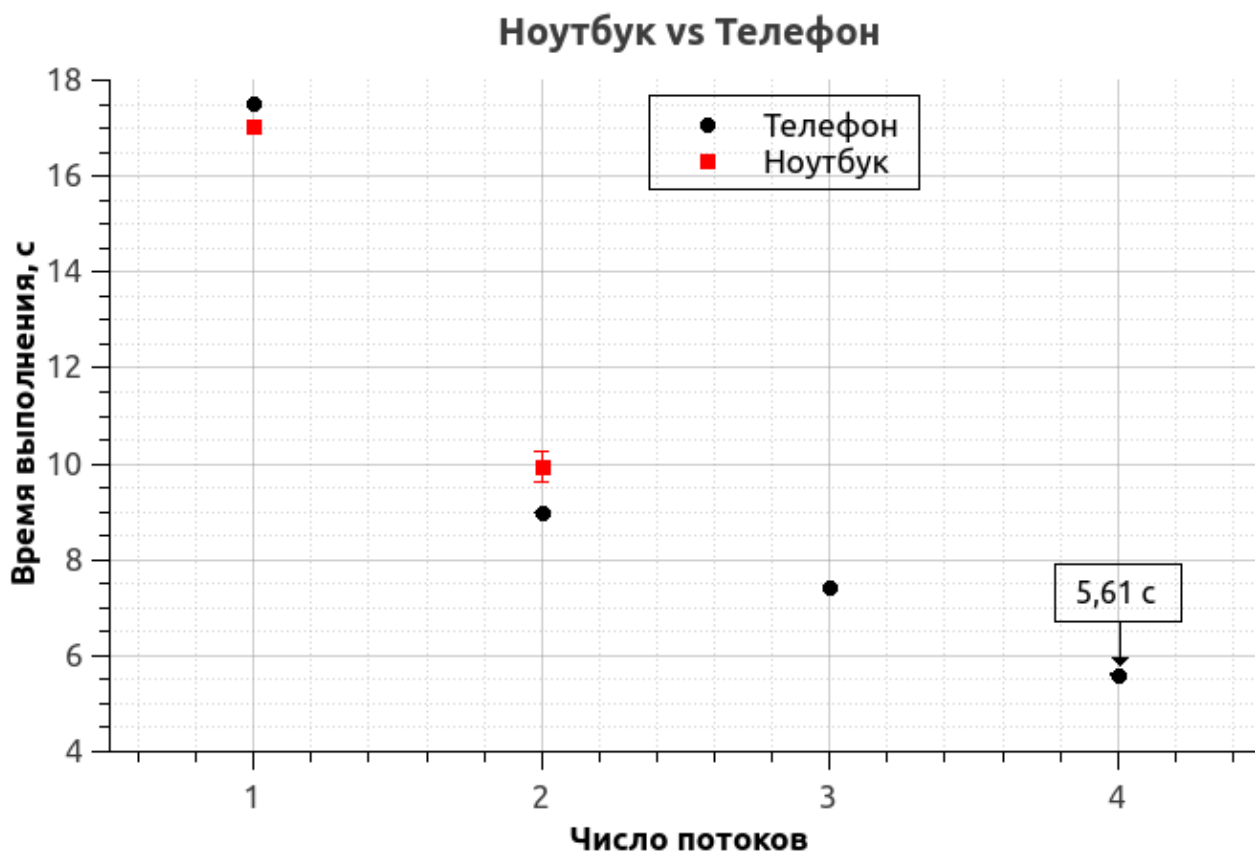


Рис. 7. Запуск на телефоне и ноутбуке.

Результат теста программы на учебном сервере (с 8-ядерным процессором Intel Xeon X5660) представлен на рисунке 8.

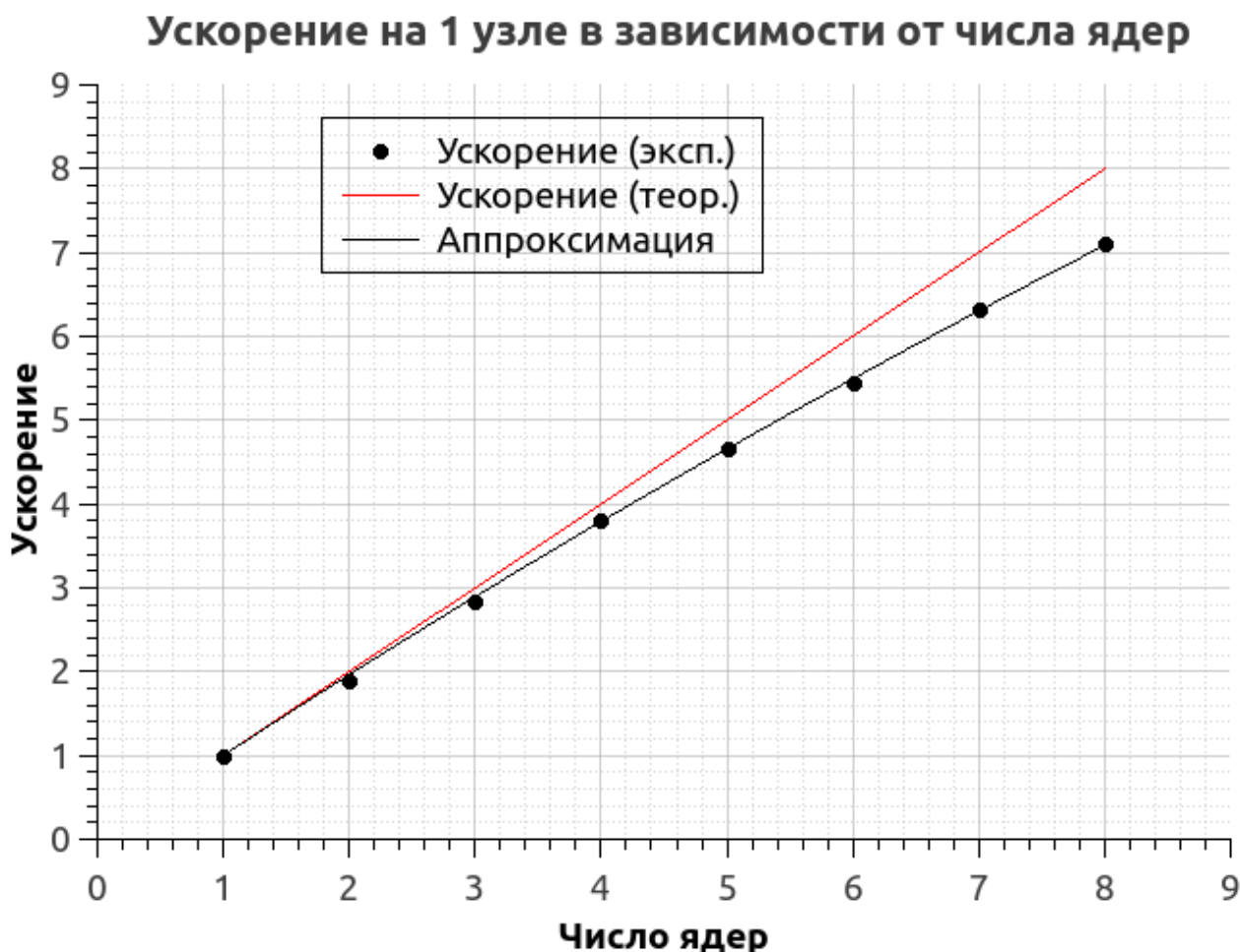


Рис. 8. Запуск на учебном сервере

При запуске MPI было указано запускать 1 экземпляр процесса, поэтому на графике ускорение только за счёт OpenMP. Красная линия – Линейная аппроксимация (соответствует идеальному параллельному алгоритму). Чёрная линия – аппроксимация в соответствии с законом Амдала. Отсюда доля последовательной части программы – около 18%.

Для анализа работы MPI использовался СК «Ломоносов» [6]. Так же были изменены параметры расчётов: количество вычислений было специально увеличено в несколько раз за счёт количества точек, в которых вычисляется поле, и уменьшения разбиения. По сложности вычисления близки к практически значимым. (Рис. 9)

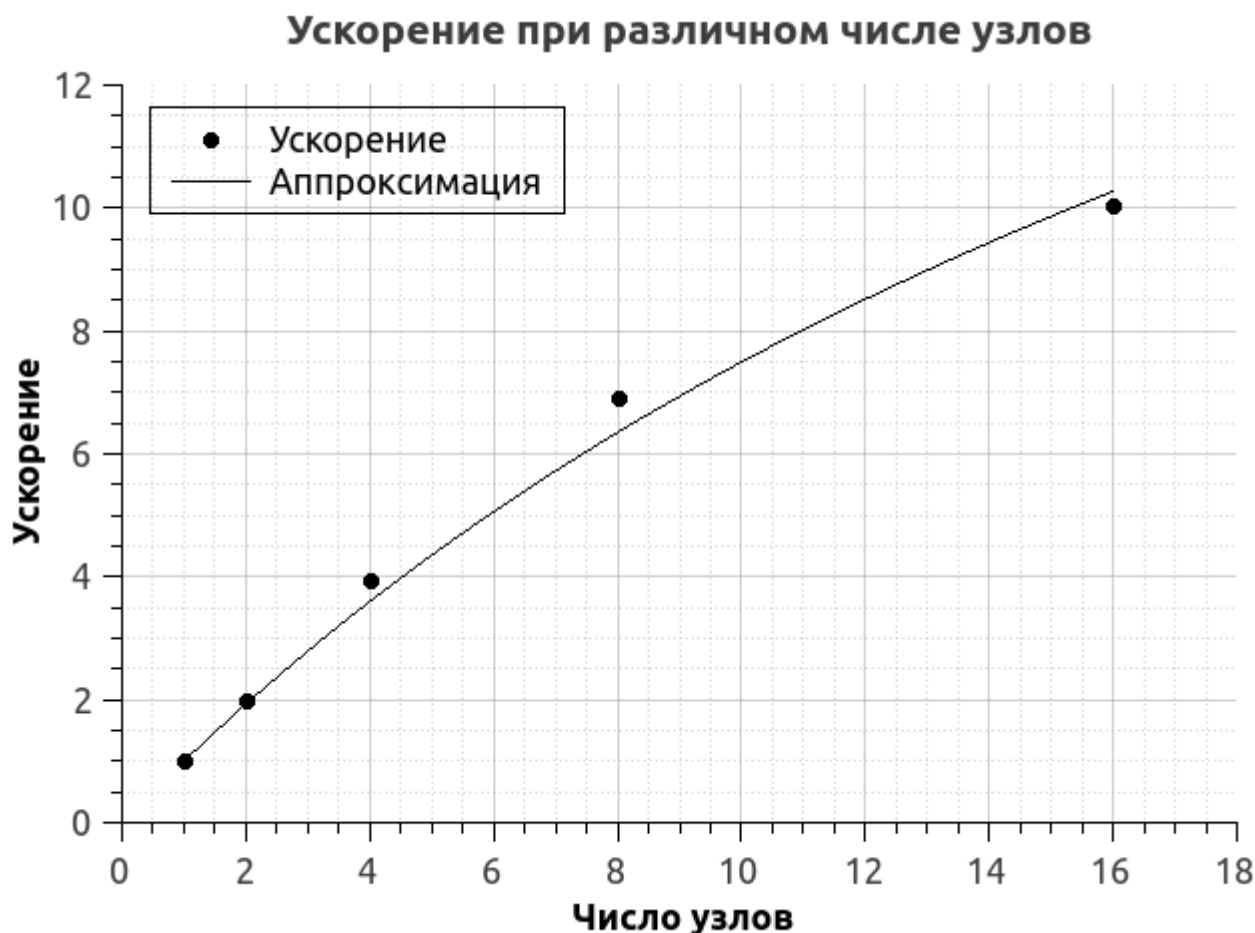


Рис. 9. Запуск на СК «Ломоносов»

Ввиду очередей и ограничений в использовании суперкомпьютера при построении графика использовалось меньше точек. Аппроксимация законом Амдала даёт долю последовательной части программы  $3,7 \pm 0,4\%$ . На 16 ядрах и 16 узлах было достигнуто наилучшее время вычисления – 2,5 секунды. Для сравнения на 1 узле программа выполнялась почти полминуты, а на домашнем компьютере расчёт длился бы несколько суток (не проверялось в целях экономии ресурсов).

#### Поиск оптимального разбиения.

Для экспериментального выявления зависимости оптимального диаметра разбиения были написаны скрипты на bash, которые автоматически задавали различные параметры фокусировки и разбиение, варьировали их и собирали данные в файлы для последующего анализа. Для анализа данных была написана вспомогательная программа, которая сравнивала распределение при исследуемом разбиении с эталонным распределением (таковым считалось распределение, полученное при диаметре разбиения 3 мкм). Выход программы – координаты центра артефактного кольца (здесь и далее артефактным будет называться максимум, вызванный отражением частот, а не физическими

явлениями) и значение амплитуды. Для анализа использовались только координаты центра (на рис. 5 это  $x_p=324$  мкм).

В результате получены следующие зависимости:

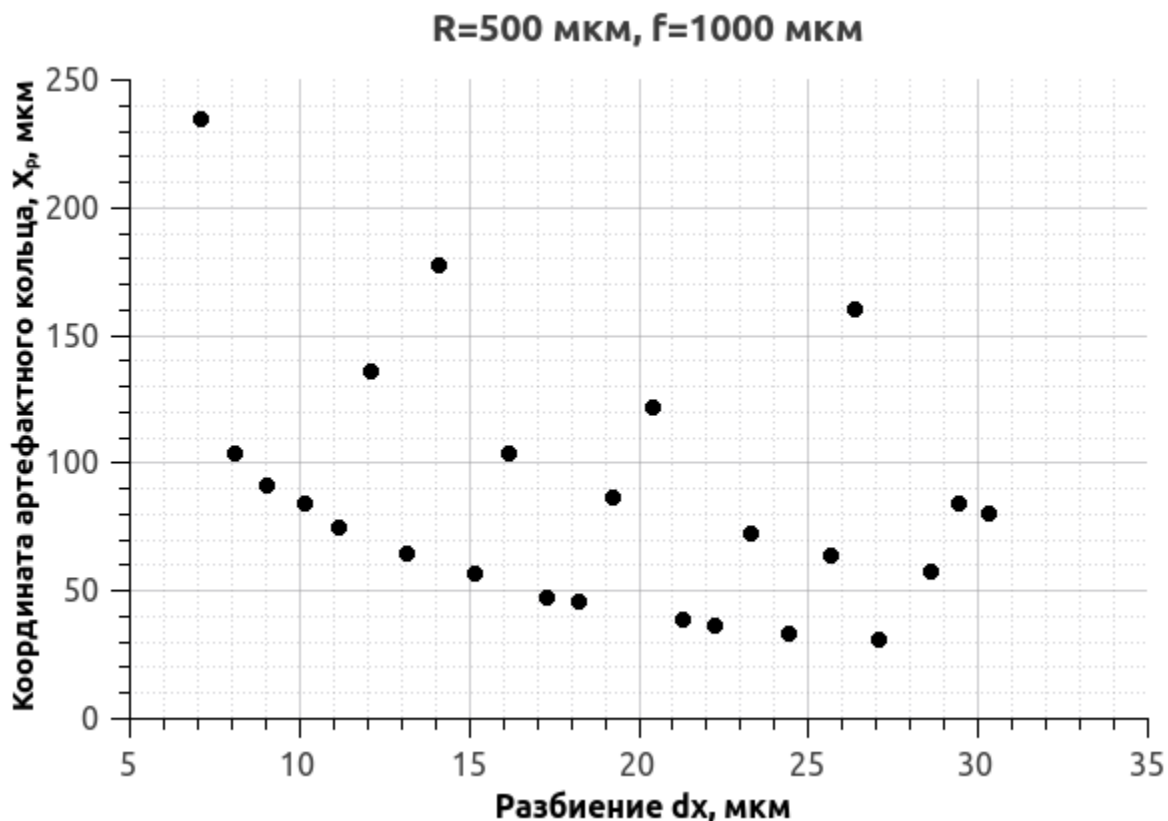


Рис. 10. Зависимость координаты артефактного максимума от шага разбиения

На графике (рис. 10) видно семейство кривых, похожих на зависимость типа  $dx=k/x^\alpha$ . Кривых получилось несколько, т. к. программа-анализатор искала наибольший максимум, а по мере ухудшения разбиения их оказывалось больше одного и второй или третий максимум могли оказаться больше первого (рис. 11). Интерес представляет только первый артефактный пик, поэтому дальнейшему анализу подверглась только нижняя кривая. Аппроксимация (рис. 12) функцией  $dx=k/x^\alpha$  показала с высокой степенью точности, что  $\alpha=1$  (с точностью до 2%), что согласуется с теорией.

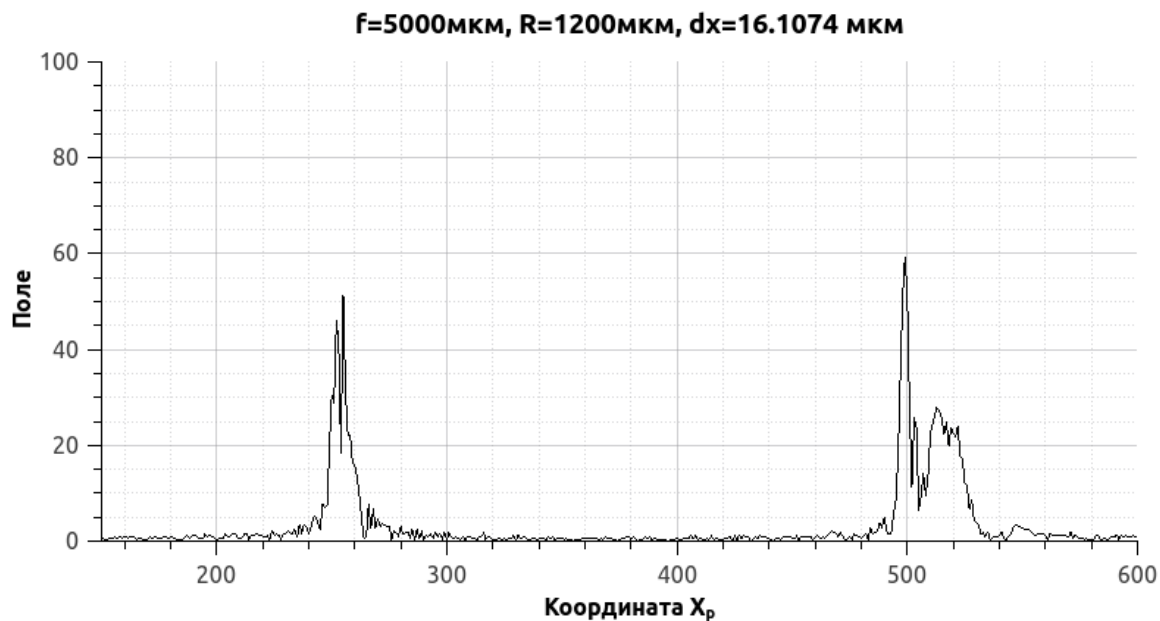


Рис. 12. Два артефактных максимума

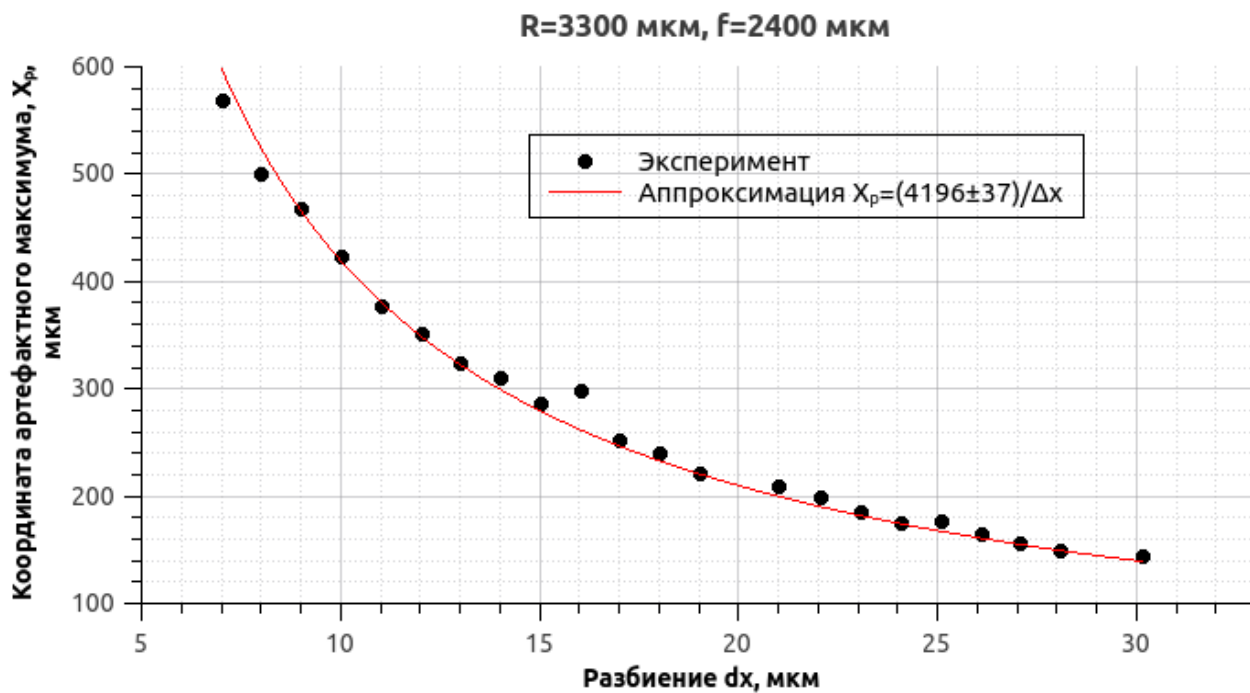


Рис. 12. Аппроксимация  $dx(x)$

Аналогично проведён анализ для фиксированных радиуса и диаметра разбиения (рис. 13).



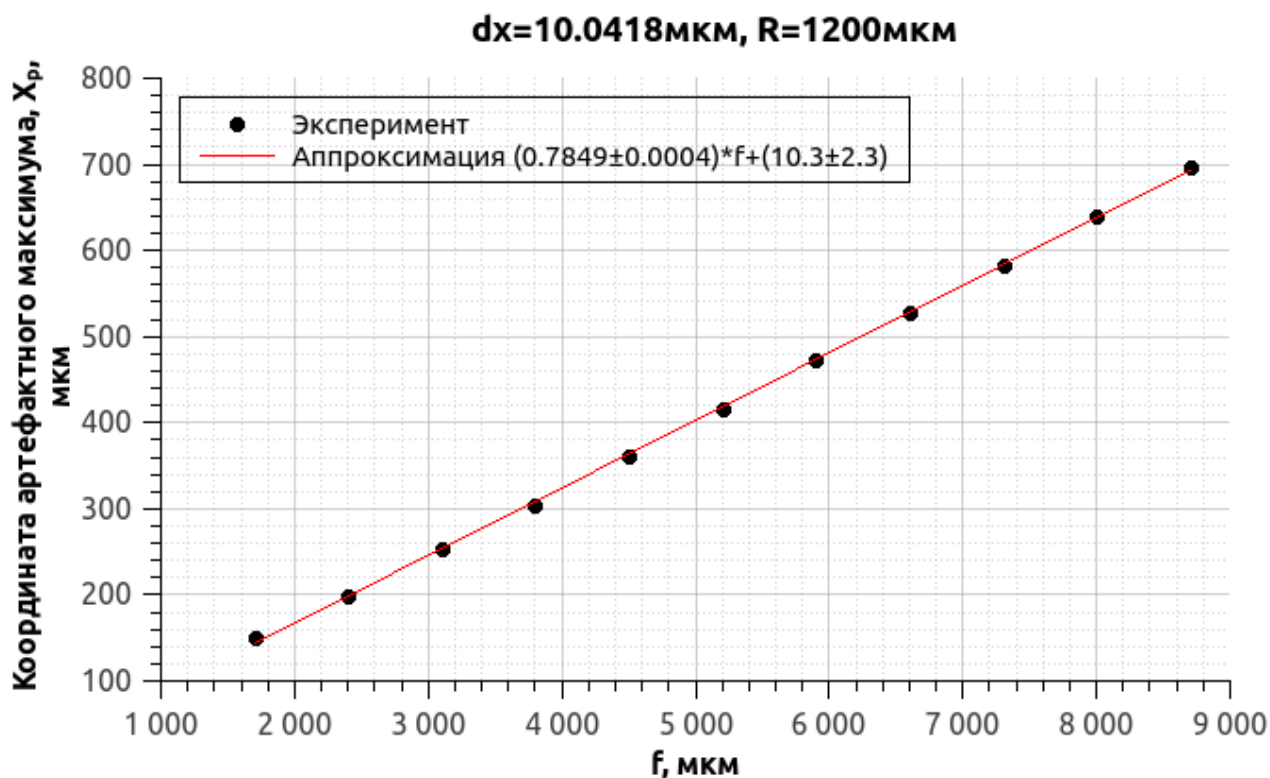


Рис. 13. Зависимость координаты артефактного максимума от фокусного расстояния

Из графика видно, что точки хорошо ( $R^2=0,99975$ ) аппроксимируются линейной зависимостью. Это тоже согласуется с теорией.

## Выводы.

В ходе работы была написана параллельная программа расчёта напряжённости электрического поля в области фокуса параболического зеркала для параллельного лазерного пучка. Код программы предполагает различные модификации и расширения для более общего случая. Программа была протестирована на различных вычислительных устройствах в параллельном режиме. При этом наблюдается заметное ускорение при распараллеливании, согласующееся с законом Амдала.

Артефактный максимум, наблюдаемый при большом шаге разбиения, объясняется отражением частот при дискретизации быстроосциллирующей подынтегральной функции. Аналитически и в расчётах показано, что координата артефактного максимума  $x_p \sim f/(k\Delta x)$ , где  $f$  – фокусное расстояние зеркала,  $k$  – волновое число,  $\Delta x$  – шаг разбиения. С помощью найденной зависимости можно оптимизировать расчетную сетку для векторных дифракционных интегралов в зависимости от "области наблюдения", в которой необходимо получить значение поля.

## Список литературы.

1. P. Varga and P. Török, «Focusing of electromagnetic waves by paraboloid mirrors. I. Theory», OSA, 2000
2. Tae Moon Jeong, Stefan Weber, Bruno Le Garrec, Daniele Margarone, Tomas Mocek and Georg Korn, «Spatio-temporal modification of femtosecond focal spot under tight focusing condition», OSA, 2015
3. <https://www.wikipedia.org/>
4. Алешкевич В.А., «Курс общей физики. Оптика», ФИЗМАТЛИТ, 2010
5. Калиткин Н.Н. «Численные методы», «Наука», 1978
6. <http://users.parallel.ru/wiki>